数据库复习

选择题

1. **修改数据表结构的语句：**
   - 添加列：`ALTER TABLE table_name ADD column_name datatype;`
   - 删除列：`ALTER TABLE table_name DROP COLUMN column_name;`
   - 修改列的数据类型：`ALTER TABLE table_name MODIFY column_name datatype;`
2. **修改数据表中数据的语句：**
   - 更新数据：`UPDATE table_name SET column1 = value1, column2 = value2 WHERE condition;`
3. **视图的定义：** 视图是虚拟的表，是基于 SQL 查询结果的可视化表，存储的是查询的 SQL 语句。创建视图：`CREATE VIEW view_name AS SELECT column1, column2 FROM table_name WHERE condition;`
4. **建立索引的目的：** 索引用于加快数据库查询速度，它类似于书籍的目录，可以快速定位到数据。通过索引，可以更快地执行检索、排序和聚合操作。
5. **用户授权语句：** 授权给用户：`GRANT privileges ON database_name.table_name TO 'username'@'host';`
6. **收回授权语句：** 收回用户权限：`REVOKE privileges ON database_name.table_name FROM 'username'@'host';`
7. **MySQL 日志文件的类型：** MySQL 有多种日志文件，包括错误日志、查询日志、慢查询日志、二进制日志和事务日志等，用于记录数据库的操作和状态。
8. **MySQL 角色的定义：** MySQL 中的角色是一组权限的集合，可以将角色授予用户，简化权限管理。
9. **规范化主要的理论依据：** 数据库规范化的主要目的是减少数据冗余，提高数据存储的效率和一致性。它基于关系数据库理论的范式概念。
10. **规范化过程主要为克服数据库逻辑结构中哪些问题：** 主要为克服数据冗余、数据插入异常、更新异常和删除异常等问题。

11. **字符串模式匹配中的通配符：** 在数据库中常见的通配符有 `%`（匹配任意字符序列）和 `_`（匹配任意单个字符）。

12. **创建存储过程的语句：** 创建存储过程：`CREATE PROCEDURE procedure_name AS ...`

13. **如何与 NULL 值进行比较：** 可以使用 `IS NULL` 或 `IS NOT NULL` 来比较是否为 NULL。

14. **同一个关系模型的任两个元组值能否相同：** 不可以，关系模型中每个元组是唯一的，不会有两个元组值完全相同。

15. **常用数据类型：** 常见的数据类型包括 VARCHAR、INTEGER、DATE、DECIMAL、FLOAT、BOOLEAN 等，用于存储不同类型的数据。

16. **数据库备份的类型：** 数据库备份的类型包括完全备份、增量备份和差异备份。

17. **三大范式的基本概念：** 第一范式（1 NF）：表中的每一列都是不可分割的原子数据项。 第二范式（2 NF）：要求表必须符合 1 NF，且所有非主属性必须完全依赖于主码。 第三范式（3 NF）：要求表必须符合 2 NF，并且不存在传递依赖。

18. **模糊查找语句：** 使用 `LIKE` 关键字进行模糊查询，配合通配符 `%`。

19. **事务的特征有哪些：** ACID 特性，即原子性、一致性、隔离性和持久性。

20. **E-R 模型向关系模型转换的时候，一个 M: N 联系转为关系模式时，该关系模式的关键字是 M 端实体关键字和 N 端实体关键字的组合：** 是的，M: N 联系转为关系模式时，通常需要创建一个新的关系表，其主键由 M 端实体关键字和 N 端实体关键字的组合构成。

判断题

22. **数据操纵语言有哪些：** 数据操纵语言（Data Manipulation Language，DML）用于对数据库中的数据进行操作。常见的 DML 包括：

    - **SELECT：** 从数据库中检索数据。
    - **INSERT：** 向数据库中插入新的数据。
    - **UPDATE：** 更新数据库中的数据。
    - **DELETE：** 从数据库中删除数据。

23. **聚合函数、视图的作用：**

- **聚合函数：** 用于对一组数据执行计算，并返回单个值。如 SUM、AVG、COUNT 等。它们可以对数据进行汇总、统计等操作。
- **视图：** 视图是虚拟的表，是基于查询结果的可视化表。它简化了复杂查询的操作，并且可以控制用户对数据的访问权限。

24. **索引的作用：** 索引是对数据库表中一列或多列的值进行排序的数据结构，类似于书籍的目录。它加快了数据库的检索速度，减少了数据查询的时间。

25. **事务原子性的定义：** 原子性是事务的 ACID 特性之一，指的是事务中的所有操作要么全部完成，要么全部不完成。事务原子性保证了数据库在执行事务时的完整性和一致性。

26. **读脏数据的定义：** 读脏数据指的是一个事务读取了另一个未提交事务中的数据，如果未提交的事务后来回滚了，则读取的数据实际上是无效的数据。

27. **死锁发生的原因：** 死锁是指两个或多个事务永久地互相等待对方持有的锁资源。它通常由事务争夺资源（如表、行）并以不同的顺序获取锁资源而引起。

28. **MySQL 中获取当前日期时间的方法：** 使用 `NOW()` 函数可以获取当前日期和时间，例如：`SELECT NOW();`

29. **数据表添加数据的方法：** 使用 `INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);` 可以向数据表中添加数据。

30. **排它锁和共享锁：**
    - **排他锁（Exclusive Lock）：** 一次只允许一个事务获得对同一资源的排他锁，用于写入或修改操作，其他事务无法获取该锁。
    - **共享锁（Shared Lock）：** 允许多个事务同时获得对同一资源的共享锁，用于读取操作，多个事务可以同时持有共享锁。

31. **主键约束的作用：** 主键约束用于唯一标识数据库表中的每一行数据，确保每行的唯一性和非空性。通常用于唯一标识数据行和建立表间关系。

32. **视图中数据和基本表数据的关系：** 视图本身不存储实际数据，而是基于查询结果生成的虚拟表。视图中的数据实际上来自于基本表，它们是相互关联的，视图可以对基本表的数据进行过滤、聚合或者进行部分数据的展示。

33. **去掉查询重复数据的方法：** 使用 `DISTINCT` 关键字可以去除查询结果中的重复数据，例如：`SELECT DISTINCT column_name FROM table_name;`

34. **丢失修改的定义：** 丢失修改指的是两个事务同时修改同一数据，其中一个事务

的修改可能被另一个事务覆盖，导致一个事务的修改丢失。

35. **存储过程的定义：** 存储过程是一系列 SQL 语句的集合，经过编译并存储在数据库中，可以像函数一样被调用执行，用于完成特定的任务并返回结果。

填空题

36. **两个实体型之间的联系的三种类型：** 在 E-R 模型中，两个实体型之间的联系可以是：
    - **一对一关系（One-to-One）：** 一个实体实例仅与另一个实体实例相关联。
    - **一对多关系（One-to-Many）：** 一个实体实例关联到多个另一个实体实例。
    - **多对多关系（Many-to-Many）：** 多个实体实例关联到多个另一个实体实例。
37. **关系型数据库中二维表的名称：** 关系型数据库中的二维表通常被称为关系（Relation）或者关系表（Relation Table）。
38. **关系模型中 3 类完整性约束：** 关系模型中的三类完整性约束包括：
    - **实体完整性（Entity Integrity）：** 确保表中的每一行都有一个唯一的标识符，一般由主键实现。
    - **参照完整性（Referential Integrity）：** 确保在引用关系中，每个外键值都是存在于被引用表的主键值上的。
    - **用户定义的完整性（User-defined Integrity）：** 由用户定义的其他约束，例如域完整性、用户自定义的触发器等。
39. **事务的定义：** 事务是数据库操作的基本单位，是一系列数据库操作（例如 SELECT、INSERT、UPDATE、DELETE）的集合。它要么全部执行，要么全部不执行。
40. **死锁的定义：** 死锁指的是两个或多个事务相互等待对方所持有的资源，导致它们都无法继续执行，从而陷入无限期等待的状态。
41. **MySQL 中封锁的两种类型、两种粒度：** 在 MySQL 中，封锁有两种类型：
    - **共享锁（Shared Lock）：** 用于读取操作，多个事务可以同时持有。
    - **排他锁（Exclusive Lock）：** 用于写入或修改操作，只允许一个事务持有。

    MySQL 中的封锁粒度包括表级封锁和行级封锁。表级封锁是对整个表进行锁定，而行级封锁是对单行数据进行锁定。

42. **E-R 图的基本成分：** E-R 图包括三种基本成分：
    - **实体（Entity）：** 数据库中可以识别的对象，通常用矩形表示。
    - **属性（Attribute）：** 实体的特征或特性，用椭圆形表示。
    - **联系（Relationship）：** 实体之间的关联，用菱形表示。
43. **视图和虚拟表的概念：**
    - **视图（View）：** 是一个基于查询结果的虚拟表，存储的是查询语句而不是实际数据，用于简化复杂查询和控制用户访问权限。
    - **虚拟表（Virtual Table）：** 通常指的是内存中的表，也可以是基于查询结果的表，但与视图不同，它通常不存储查询语句，而是存储实际数据。
44. **E-R 模型和关系模型的对应关系：** E-R 模型是用于概念设计的数据模型，通过实体、属性和关系描述现实世界的实体及其之间的联系。而关系模型是基于关系代数理论，通过表格（关系）描述数据模型，是 E-R 模型的一种具体实现方式。
45. **删除数据表的 SQL 指令：** 使用 `DROP TABLE table_name;` 可以删除数据表。
46. **输出数据表中数据的 SQL 指令：** 使用 `SELECT * FROM table_name;` 可以输出数据表中的所有数据。
47. **限制查询行数的 SQL 语句：** 使用 `LIMIT` 关键字可以限制查询结果的行数，例如：`SELECT * FROM table_name LIMIT 10;` 将返回前 10 行数据。
48. **授权指令：** 授权指令用于赋予用户或角色相应的权限。例如，`GRANT SELECT ON table_name TO 'user'@'localhost';` 授予用户对表的查询权限。
49. **PRIMARY KEY（主键）、FOREIGN KEY（外键）的定义：**
    - **主键（PRIMARY KEY）：** 用于唯一标识表中的每一行数据，确保其唯一性和非空性。
    - **外键（FOREIGN KEY）：** 用于建立表与表之间的关系，确保参照表中的外键值存在于被引用表的主键中。

问答题

当涉及 SQL 代码编写时，以下是一些基本的语句示例：

创建表

```SQL
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Age INT,
    GPA DECIMAL(3, 2)
);
```

数据表的增加

```SQL
INSERT INTO Students (StudentID, FirstName, LastName, Age,
GPA)
VALUES (1, 'John', 'Doe', 20, 3.5);

INSERT INTO Students (StudentID, FirstName, LastName, Age,
GPA)
VALUES (2, 'Jane', 'Smith', 22, 3.8);
```

数据表的删除

```SQL
DELETE FROM Students WHERE StudentID = 1;
```

数据表的修改

```SQL
UPDATE Students SET Age = 23 WHERE StudentID = 2;
```

数据表的查询

```SQL
SELECT * FROM Students;
```

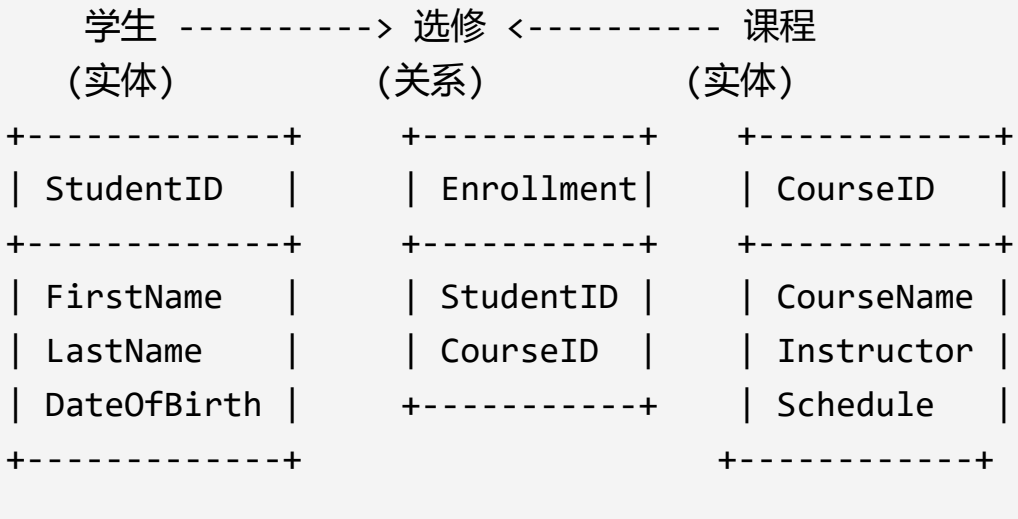
这些代码示例演示了如何创建一个简单的学生表、向表中添加数据、删除特定行、修改数据以及检索整个表的数据。实际操作可能需要更多的列和条件，这里仅供参考。

综合题

E-R 图的传统方式通常使用特定形状来表示不同的元素，例如矩形表示实体、菱形表示关系、椭圆形表示属性。下面是一个简单的例子：

假设有两个实体：学生（Student）和课程（Course），它们之间存在多对多的关系，一个学生可以选修多门课程，一门课程也可以被多个学生选修。

E-R 图绘制



- **学生（Student）** 和 **课程（Course）** 是实体，用矩形表示。
- **选修（Enrollment）** 是关系，用菱形表示。
- 每个实体的属性使用椭圆形表示，如学生的属性包括 StudentID、FirstName、LastName、DateOfBirth，课程的属性包括 CourseID、CourseName、Instructor、Schedule。

关系模型文字描述

学生表（Students）

- 主键：StudentID

StudentID (PK)	FirstName	LastName	DateOfBirth
1	John	Doe	1995-05-15
2	Jane	Smith	1994-08-21

课程表（Courses）

- 主键：CourseID

CourseID (PK)	CourseName	Instructor	Schedule
101	Mathematics	Prof. Lee	Mon/Wed/Fri
102	Physics	Prof. Chen	Tue/Thu

选修关系表（Enrollments）

- 复合主键：StudentID, CourseID
- 外键：StudentID (指向学生表中的主键), CourseID (指向课程表中的主键)

StudentID (PK, FK)	CourseID (PK, FK)
1	101
1	102
2	102

这里，Enrollments 表中的主键由 StudentID 和 CourseID 组成，分别是对学生表和课程表的外键。这样可以建立学生和课程之间的多对多关系。